

## MI 2022

1. Upišite niz znakova koji nedostaje (#####) u navedenom CSS kodu?

```
#header h1 {  
  animation-duration: 2s;  
  animation-name: ImeAnimacije;  
  animation-iteration-count: 10;  
  animation-timing-function: linear;  
  animation-direction: alternate;  
}  
  
##### ImeAnimacije {  
  from {  
    margin-left: 100%;  
  }  
  to {  
    margin-left: -100%;  
  }  
}
```

**Odgovor:** @keyframes

2. Zaštićenoj stranici namijenjenoj samo prijavljenim korisnicima pokušao je pristupiti anonimni klijent. Koji HTTP status bi trebao dobiti kao odgovor? (Ne koristi se Form-based authentication)

**Odgovor:** 401

3. Čemu služi pristupna točka /.well-known/openid-configuration na OIDC serveru?

**Odgovor:** ZA INFORMACIJU O ADRESI PRISTUPNE TOČKE KOJA ĆE VRATITI DETALJNE INFORMACIJE O KORISNIKU, TJ. VLASNIKU NEKOG TOKENA ;  
ZA INFORMACIJU O POPISU PODRŽANIH OPSEGA ;  
ZA INFORMACIJU O ADRESI NA KOJOJ SE MOGU PREUZETI JAVNI KLJUČEVI KOJIMA SE PROVJERAVA TOKEN

4. Ako se na poslužitelju ne koriste sigurne reference na objekte pa je osim pristupa putanji /user/getAccounts logiranom korisniku bez admin ovlasti dozvoljen pristup putanji: /admin/getAccounts čime korisnik dobiva admin ovlasti, o kojem sigurnosnom nedostatku se radi?

**Odgovor:** LOŠA KONTROLA PRISTUPA

5. Koja strategija se preporuča za otklanjanje ranjivosti loša kontrola pristupa?

**Odgovor:** ZAMJENA JAVNO DOSTUPNIH INTERNIH REFERENCI S PRIVREMENIM VRIJEDNOSTIMA KOJE SE NA POSLUŽITELJU PRESLIKAVAJU U PRAVE

6. Za pisanje ovog ispita prijavili ste se u Edgar koristeći [AAI@Edu.hr](mailto:AAI@Edu.hr) ili neku drugu vanjsku uslugu. U kontekstu korištenja usluge za autentifikaciju, što ste vi? **Odgovor:** RESOURCE OWNER

7. Web aplikacija koristi sljedeći kod za unos podataka od korisnika:

```
<form action="search.php" method="GET" />
  Dobrodošli!
  <p>Upišite svoje ime: <input type="text" name="ime" /><br />
  <input type="submit" value="Go" /></p><br>
</form>
<?php
  echo "<p>Vaše ime je:<br />";
  echo ($_GET[ime]);
?>
```

Napadač u input polje **ime** unosi:

[http://ranjiva\\_stranica/clean.php?ime=<script>alert\(document.cookie\);</script>](http://ranjiva_stranica/clean.php?ime=<script>alert(document.cookie);</script>)

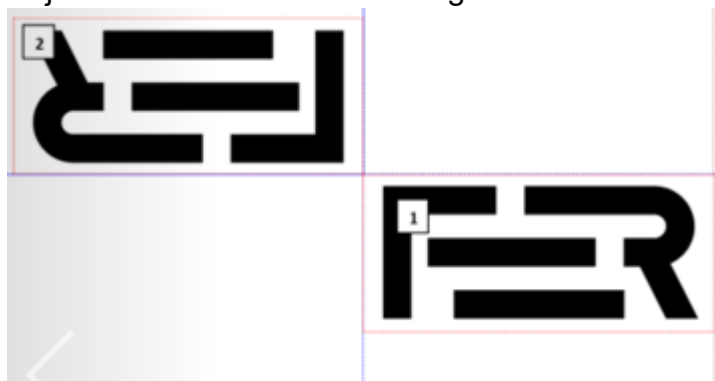
Koju vrstu sigurnosnog nedostatka ovaj napad koristi?

**Odgovor: XSS SIGURNOSNI NEDOSTATAK**

8. Što je istinito:

- a) Web aplikacije razvijene u sveobuhvatnim radnim okvirima (component frameworks, npr. Bootstrap) imaju nedostatak da „sve izgledaju isto“.
- b) Obični („vanilla“) CSS nam omogućuje punu kontrolu nad izgledom sučelja
- c) Obični („vanilla“) CSS nam omogućuje brži razvoj nego da koristimo radne okvire
- d) Kod običnog („vanilla“) CSS-a i klasičnog semantic CSS pristupa tipično moramo pisati sav CSS kod ispočetka („from scratch“) odnosno ne možemo ga ponovo iskoristiti u većoj mjeri za druge projekte
- e) Radni okviri s pomoćnim klasama (utility frameworks, npr. Tailwind) nam omogućuju da slijedimo dobre prakse
- f) Web aplikacije razvijene u radnim okvirima s pomoćnim klasama (utility frameworks, npr. Tailwind) imaju nedostatak da „sve izgledaju isto“.
- g) Radni okviri s pomoćnim klasama proizvode minimalni CSS, te štedimo na veličini CSS datoteka odnosno mrežnom prometu
- h) Razvoj pomoću radnih okvira (Tailwind, Bootstrap) tipično donosi nepotrebnii višak programskog koda (overhead code)

9. Kojim CSS izrazom možemo logo FER-a iz 1 u 2?



**Odgovor: transform: translate(-100%, 100%) rotate (180deg);**

10. Kako se naziva mehanizam u kojem je jedna HTTP/2 poruka podijeljena na međusobno neovisne okvire koji se mogu asinkrono slati i primiti jednom TCP konekcijom? **Odgovor: MULTIPLEKSIRANJE ZAHTEJEVA**

11. Izbaci uljeza: Identity provider, Resource owner, **Proxy server**, Resource server, Relying party

12. Što je to salting poruke? **Odgovor: POSTUPAK UMETANJA NIZA POZNATIH ZNAKOVA DOGOVORENE DUŽINE U ZAPORKU PRIJE HASHIRANJA**

13. Zaglavlje kojim se šalje token poslužitelju ima oblik:  
**Odgovor: Authorization: Bearer sadržaj\_tokena**

14. Pristupni token kod protokola OAuth2 sadrži popis dozvola kojima korisnik raspolaže. Taj skup dozvola se u terminologiji protokola OAuth2 naziva:  
**Odgovor: SCOPES**

15. Unesi CSS koji će nastati prevođenjem sljedećeg SAAS (.scss) koda:

```
ul {  
  li {  
    font: {  
      family: sans-serif;  
      size: 10px;  
    }  
  }  
}
```

**Odgovor: ul li {  
font-family: sans-serif; font-size: 10px;  
}**

16. Kako ćemo, slijedeći BEM metodologiju, postaviti crvenu boju teksta naslova gumba?

**e**

```
.button--title {  
  color: red;  
}  
...  
<button>  
  <span class="button--title">Button title</span>  
</button>
```

17. Zamislamo da Edgar ima funkcionalnost kojom bi bodove svih studenata ostvarene na ovom ispitu kopirao u FER-web pozivom odgovarajućeg servisa. Koji OAuth2/OIDC tok bi se koristio? **Odgovor: Client Credentials Flow**

18. Označite sve točne tvrdnje vezane za XML External Entity(XEE) sigurnosni nedostatak:

- a) U otklanjanju ovog nedostatka potrebno je ograničiti veličinu učitane XML datoteke
- b) Učitavanje vanjskih XML datoteka je štetno i potrebno ga je onemogućiti
- c) Ranjive su sve aplikacije koje parsiraju XML datoteke
- d) Potrebno je proučiti i ažurirati postavke XML parsera za učitavanje vanjskih entiteta
- e) Potrebno je izbjegavati korištenje složenijih XML struktura ako nisu potrebne

19. Što je cilj napadača ako u SQL umetanju koristi ulančane upite?

**Odgovor: POTPUNO OVLADAVANJE I MANIPULACIJA BAZOM PODATAKA**

20. Koja strategija se preporuča za otklanjanje ranjivosti lažiranja zahtjeva na drugom sjedištu?

- a) Sanitizacija unosa od strane korisnika
- b) Filtriranje IP adresa
- c) Koristiti HTTP POST umjesto HTTP GET
- d) Pohrana tokena u korisničkoj sesiji i dodavanje tokena u sve obrasce i linkove
- e) Eliminacija izravnih referenci s privremenim neizravnim vrijednostima
- f) Inverzni Turingov test (CAPTCHA)

21. Pretpostavimo da dvije aplikacije (X i Y) koriste uslugu [AAI@Edu.hr](mailto:AAI@Edu.hr) i protokole OAuth2/OIDC. Prilikom prijave u aplikaciju X i uspješnog unosa korisničkog imena i lozinke na usluzi AAI, prijavljeni smo u aplikaciji X. Nakon toga u istom pregledniku posjetimo aplikaciju Y te se želimo prijaviti u njoj. Što od navedenog je istina? **Odgovor: APLIKACIJA Y ĆE NAS PREUSMJERITI NA USLUGU [AAI@EDU.HR](mailto:AAI@EDU.HR), ALI ĆEMO ODMAH BITI PREUSMJERENI NATRAG I BITI PRIJAVLJENI NA APLIKACIJU Y KAO POSLJEDICU ČINJENICE DA SMO OD RANIJE IMALI VALJANI COOKIE ZA AAI.**

22. Web aplikacija za autentifikaciju koristi sljedeći SQL kod:

```
String SQLQuery = "SELECT Username FROM Users WHERE Username = '" + username + "' AND Password = '" + password + "'";
```

Koje nizove znakova napadač mora unijeti za vrijednost varijabli username i password kako bi se mogao prijaviti na sustav kao prvi korisnik u korisničkoj tablici Users?

- a) ' OR '=' i ' OR '='
- b) '\*' i '\*'
- c) ' OR '=' i "
- d) ' OR '=' i '\*'
- e) ' OR '=' i ';

23. Označite sve točne tvrdnje vezane za Basic Authentication
- a) Prilikom svakog zahtjeva šalje se korisničko ime i lozinka
  - b) Lozinka je kriptirana asimetričnim ključem (privatni i javni)
  - c) Za kodiranje lozinke koristi se Base64
  - d) Prilikom svakog zahtjeva šalje se korisničko ime i sažetak lozinke
  - e) Sažetak lozinke kodiran je u Base64 formatu

24. Za pisanje ovog ispita prijavili ste se u Edgar koristeći [AAI@Edu.hr](mailto:AAI@Edu.hr) ili neku drugu vanjsku uslugu. U kontekstu korištenja vanjske usluge za autentifikaciju, što je Edgar? **Odgovor: CLIENT**

25. Koji sigurnosni propust napadač želi iskoristiti ako koristi sljedeću proceduru napada:

1. Napadač kreira hiperlink koji, osim URL-a legitimnog poslužitelja, sadrži i zlonamjerni skriptni kod.
2. Napadač šalje zlonamjerni hiperlink korisniku (npr. putem elektroničke pošte) tako da isti izgleda kao da potječe od strane legitimnog poslužitelja.
3. Korisnik aktivira hiperlink, pri čemu se legitimnom web poslužitelju šalje HTTP zahtjev za ranjivom web stranicom. Korisnik je prijavljen na legitimni poslužitelj.
4. Legitimni web poslužitelj generira dinamičku web stranicu koja zbog ovog sigurnosnog propusta sadržava umetnuti zlonamjerni skriptni kod i šalje je korisniku kao HTTP odgovor.
5. Zlonamjerni skriptni kod iz zaprimljene web stranice izvršava se unutar korisnikovog web preglednika s istim ovlastima kao da je potekao od legitimnog poslužitelja.
6. Zlonamjerni skriptni kod dohvati korisnikove kolačiće vezane uz legitimni poslužitelj.
7. Korisnikov web preglednik dobiva informaciju o preusmjeravanju na maliciozni web poslužitelj koji je pod napadacevom kontrolom.
8. Sakupljeni kolačići šalju se na maliciozni poslužitelj bez korisnikovog znanja. Napadaču se otvara mogućnost krađe korisnikovog identiteta.

**Odgovor: JEDNOKRATNI (REFLEKTIRANI) XSS SIGURNOSNI PROPUST**

26. Upišite niz znakova koji nedostaje (####) u CSS kodu?

```
#header h1 {
  animation-duration: 12s;
  #####: ImeAnimacije;
  animation-iteration-count: 10;
  animation-timing-function: linear;
  animation-direction: alternate;
}

@keyframes ImeAnimacije {
  from {
    margin-left: 100%;
  }
  to {
    margin-left: -100%;
  }
}
```

**Odgovor: animation-name**

27. Izbacite uljeza u kontekstu pojma Single Sign On:

- a) Auth0
- b) Cookie
- c) CSRF
- d) OAuth2
- e) OIDC

28. Koja strategija se preporuča za otklanjanje ranjivosti lažiranja zahtjeva na drugom sjedištu?

- a) **Korištenje dodatne autentifikacije za osjetljive funkcije (npr. APLI)**
- b) Filtriranje IP adresa
- c) Eliminacija izravnih referenci s privremenim neizravnim vrijednostima
- d) Inverzni Turing
- e) Sanitizacija unosa od strane korisnika

29. Podaci unutar identifikacijskog tokena u OpenId Connect protokolu nazivaju se: **Odgovor: CLAIMS**

30. Zašto je razvijen test reCAPTCHA kao unaprjeđenje CAPTCHA-e?

**Odgovor: TEST CAPTCHA POSTAO JE PRETEŽAK ZA LJUDE, A PRELAGAN ZA RAČUNALA**

31. Ako web-aplikacija koristi OAuth2/OIDC i tok Authorization code flow, tada ona u nekom trenutku u pozadini komunicira s OAuth2/OIDC serverom mijenjajući

- a) Identity token za access token
- b) Access token za refresh token
- c) Token za kod
- d) **Kod za token(e)**
- e) Kod za cookie

32. Unesi CSS koji nastaje prevođenjem SASS (.scss) koda:

```
%default {  
  color: black;  
}  
%alternative {  
  color: red;  
}  
$brand-color: blue;  
  
h1 {  
  @extend %default;  
  font-size: 2rem;  
}  
  
h2 {  
  @extend %default;  
  font-size: 1rem;  
}
```

```
h2, h1 {  
  color: black;  
}  
  
h1 {  
  font-size: 2rem;  
}  
  
h2 {  
  font-size: 1rem;  
}
```

**Odgovor**

33. Pristupni token kod protokola OAuth2 sadrži popis dozvola kojima korisnik raspolaže. Taj skup dozvola se u terminologiji protokola OAuth2 naziva:

- a) **Scopes**
- b) Roles
- c) Access identity
- d) Permissions
- e) Claims

34. Ako napadač želi saznati strukturu tablica i relacijske baze podataka koji oblik napada SQL umetanjem se za to tipično koristi?

- a) Upit Union
- b) Tautologija
- c) Slijepo umetanje
- d) **Ilegalni upiti**
- e) Ulančani upiti

35. Zamislite situaciju. Napisali ste mobilnu aplikaciju u koju se korisnici prijavljuju koristeći AAI@Eduhr, nakon čega aplikacija šalje token (kojeg je izdao AAO@EduHr) Edgaru tražeći od Edgara da vam prikaže trenutni broj bodova na predmetu. U kontekstu korištenja vanjske usluge za autentifikaciju, što je AAI@EduHr?

- a) Proxy
- b) **Authorization server**
- c) Resource owner
- d) Resource server
- e) Client

36. CAPTCHA test koristi se u višefaktorskoj autentifikaciji za odgovor na koje pitanje? **Odgovor: ŠTO SAM**

37. Situacija u kojoj pozivatelj (npr. Javascript kod) i web.servis koji biva pozvan nisu na istom serveru (**portu**) predstavlja primjer onoga što se u literaturi naziva: **Odgovor: CORS = CROSS-ORIGIN RESOURCE SHARING**

38. Pretpostavimo da klijent prvo šalje HTTP zahtjev A koji zahtijeva znatne resurse poslužitelja u nakon toga HTTP zahtjev B koji ne zahtijeva znatne računalne resurse. Poslužitelj će obraditi zahtjev B brzo, ali ne može poslati odgovor jer čeka na kraj obrade zahtjeva A kako bi poštvao redosljed pristiglih zahtjeva. Na taj način spori zahtjevi postaju ograničenje svih sljedećih zahtjeva. Kako se zove ova situacija? **Odgovor: HEAD-OF-LINE-BLOCKING**

39. Kako ćemo u duhu nezavisnih CSS radnih okvira poput Tailwinda postaviti crvenu boju teksta naslova gumba?



```
.text-red {  
  color: red;  
}  
  
<button>  
  <span class="text-red">Button title</span>  
</button>
```

40. Prijavljeni korisnik pokušao je pristupiti stranici kojoj nema pravo. Koji HTTP status je trebao dobiti kao odgovor? **Odgovor: 403**

41. Podaci korisnika u identifikacijskom tokenu u OpenId Connect protokolu podijeljeni su u grupe koje se nazivaju: **Odgovor: SCOPES**

42. Ako napadač želi dohvatiti nazive svih tablica u relacijskoj bazi podataka koji oblik napada SQL umetanjem se za to tipično koristi? **Odgovor: UPIT UNION**

43. Označite sve točne tvrdnje vezane za Digest Authentication:

- a) Lozinka je kriptirana asimetričnim ključem (privatni i javni)
- b) Prilikom svakog zahtjeva šalje se korisničko ime i sažetak nastao temeljem lozinke i parametara koje je server isporučio prilikom odbijanja inicijalnog zahtjeva s informacijom o načinu provjere autentičnosti
- c) Za kodiranje lozinke koristi se Base64
- d) Prilikom svakog zahtjeva šalje se korisničko ime i lozinka
- e) Sažetak se računa i na klijentu (pregledniku) i na serveru i sažeci moraju biti isti.

44. Zamislite situaciju. Napisali ste mobilnu aplikaciju u koju se korisnici prijavljuju koristeći AAI@Eduhr, nakon čega aplikacija šalje token (kojeg je izdao AAO@EduHr) Edgaru tražeći od Edgara da vam prikaže trenutni broj bodova na predmetu. U kontekstu korištenja vanjske usluge za autentifikaciju, što je Edgar? **Odgovor: Resource server**

45. Koji sigurnosni propust se događa ako web aplikacija pročita podatke koje je poslao korisnik i pošalje ih u web preglednik bez prethodne provjere valjanosti i čišćenja umetnutog koda? **Odgovor: XSS SIGURNOSNI NEDOSTATAK**



46. Koji propust napadač želi iskoristiti ako koristi sljedeću proceduru napada:

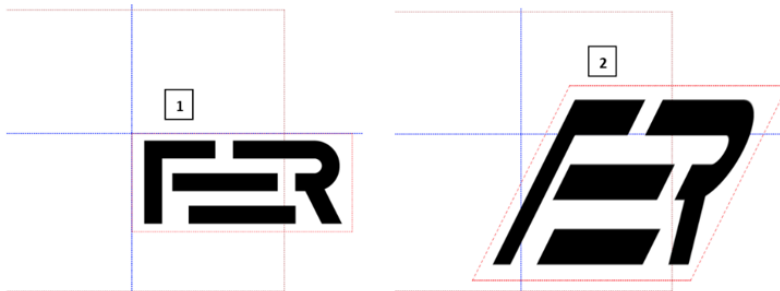
1. Napadač kreira hiperlink koji osim URL-a legitimnog poslužitelja sadrži i zlonamjerni skriptni kod
2. Napadač šalje zlonamjerni hiperlink korisniku (putem e pošte)
3. Korisnik aktivira hiperlink, pri čemu se legitimnom web poslužitelju koji sadrži sigurnosni propust šalje HTTP zahtjev za ranjivom web stranicom
4. Legitimni web poslužitelj šalje korisniku ranjivu web stranicu kao HTTP odgovor. Zlonamjerni skriptni kod nije umetnut u poslanu web stranicu, nego je još uvijek samo unutar hiperlinka
5. Korisnikov web preglednik interpretira ranjivu web stranicu koja se sada nalazi na lokalnom korisnikovom sustavu. Nailaskom na ranjivi dio stranice, aktivira se zlonamjerni kod iz hiperlinka, koji se potom izvršava s ovlastima web preglednika unutar lokalne zone korisnikovog računala.

**Odgovor: LOKALNI (DOM) XSS SIGURNOSNI PROPUST**

47. Kako se naziva mehanizam u kojem se više HTTP zahtjeva šalje jednom TCP konekcijom bez čekanja na njihove pojedinačne odgovore? **Odgovor: HTTP PIPELINING**

48. Ako se za provjeru autentičnosti koristi mehanizam Digest Authentication korisnik se može odjaviti iz aplikacije: **Odgovor: SAMO ZATVARANJEM PREGLEDNIKA**

49. Kojim CSS izrazom možemo logo FER-a iz 1->2



**Odgovor: transform: scale(1,2) skew(-45deg,0deg);**

## ZI 2022

1. Kod izrade **skalabilnih arhitektura aplikacija** označite sve tvrdnje u vezi **načela jednostavnosti** koje su istinite:

- a) Jednostavnost se može postići korištenjem TTD (test-driven development) metodologije
- b) Potrebno je predvidjeti svaki mogući scenarij i svaki rubni uvjet
- c) Jednostavnost se ne može postići korištenjem TTD-a, štoviše potrebno ga je uvijek izbjegavati
- d) Načelo jednostavnosti temelji se na korištenju jednostavnih tehnologija na poslužiteljima
- e) Načelo jednostavnosti temelji se na načelima objektno orijentiranog programiranja i cilj je postići lokalnu jednostavnost strukture koda.
- f) Potrebno je izbjeći predviđanje svakog mogućeg scenarija i svakog rubnog uvjeta jer se time gubi fokus s najčešći scenarija

2. Ako su Milk i Beverage definirani s

```
type Milk = {  
  brand : string;  
  fat : number;  
  volume : number;  
}  
  
type Beverage = {  
  type : string;  
  brand : string;  
  volume : number;  
}
```

što će ispisati sljedeći programski odsječak

```
let x : Milk | Beverage = {  
  type : 'juice',  
  brand : 'Home made',  
  volume : 2  
};  
console.log(typeof x);  
console.log(typeof (x as Beverage));
```

**Odgovor: object  
object**

3. Koje od sljedećih izjava su točne?

- a) Prilikom iscrtavanja odnosno osvježavanja prikaza, Vue koristi tzv. **Virtualni DOM** kako bi optimirao taj proces
- b) Vue ne omogućuje dvosmjerno povezivanje (two way binding)
- c) Vue programeru izlaže svoj životni ciklus (lifecycle events) na način da postoje funkcije (lifecycle hooks) **prije i poslije** svakog od tih događaja, npr. **onBeforeMount()** i **onMounted()**
- d) Na jednoj stranici može biti samo jedna Vue aplikacija
- e) Vue nema vlastiti ugrađeni state management library, već se najčešće koristi (dodatno instalira) vanjski library koji se zove **Vuex**
- f) Vue ima vlastiti ugrađeni state management library koji se zove **Vuex**

4. Ako je u Typescriptu napisan sljedeći kod:

```
function sum(a, b) {  
    return a + b;  
}
```

Kojeg tipa je povratna vrijednost iz funkcije sum?

**Odgovor: ANY**

5. Kod izrade **skalabilnih arhitektura aplikacija** označite sve tvrdnje u vezi **načela otvoreno-zatvoreno koje su istinite**:
- a) Središnji cilj ovog načela je **potpuno onemogućiti** buduće izmjene softvera i time potpuno ukloniti sve troškove promjena
  - b) Ovo načelo odnosi se na stvaranje koda koji se **mora** mijenjati kada ase promijene zahtjevi ili kada se pojave novi obrasci uporabe
  - c) Kod je **otvoren** za proširivanje i **zatvoren** za izmjenu
  - d) Kod je **zatvoren** za proširivanje i **otvoren** za izmjenu
  - e) Ovo načelo odnosi se na stvaranje koda koji se **ne** mora mijenjati kada ase promijene zahtjevi ili kada se pojave novi obrasci uporabe
  - f) Središnji cilj ovog načela je **povećati fleksibilnost** softvera i učiniti buduće promjene **jeftinijima**
6. Kod izrade **skalabilnih arhitektura aplikacija** označite sve tvrdnje u vezi **vertikalnog skaliranja koje su istinite**:
- a) Vertikalnim skaliranjem povećavaju se performanse postojećih poslužitelja ili resursa
  - b) Vertikalnim skaliranjem dodaje se više **istih** poslužitelja ili resursa
  - c) Kod vertikalnog skaliranja **ne postoji** scale down već samo scale up
  - d) Kod vertikalnog skaliranja osim scale up postoji i scale down jer kvalitetan skalabilan sustav bi trebao omogućiti i smanjenje kapaciteta
  - e) Vertikalno skaliranje pogodno je za male web aplikacije gdje se kapacitet može održavati **povećanjem** kapaciteta postojećij poslužitelja i veličine već korištenih resursa
  - f) Vertikalno skaliranje **nije** pogodno za male web aplikacije gdje se kapacitet može održavati **povećanjem** kapaciteta postojećij poslužitelja i veličine već korištenih resursa
7. Koje od sljedećih tvrdnji vezane uz service worker su istinite?
- a) Service worker izvodi se mimo glavne preglednikove UI dretve
  - b) Service worker radi samo ako je stranica poslužena putem HTTPS-a ili preko localhosta.
  - c) Service worker ne može koristiti sinkroni localStorage API, pa zato tipično koristi asinroni IndexedDB
  - d) Kod prvog otvaranja stranice, service worker moramo **registrirati** nakon čega preglednik **dohvaća, instalira i aktivira** SW čime on preuzima kontrolu nad klijentima u svoj opseg (scope)
  - e) Service worker nema pristup DOM-u
  - f) Kada zatvorimo sve kartice/preglednike s domenom/scopem koje je neki service worker kontrolirao, onda se gasi i prestaje raditi i taj service worker

8. Koje od sljedećih tvrdnji su istinite?

- a) Dozvolom za notifikaciju, web-aplikacija dobija i implicitnu dozvolu za push notifikacije te ju ne mora posebno tražiti
- b) Push notifikacije uvelike ovise o preglednicima jer se oslanjaju na push service infrastrukturu u oblaku, koju svaki proizvođač preglednika implementira zasebno
- c) VAPID ključ nam treba kako bi od korisnika zatražili dozvolu za korištenje push notifikacija
- d) Notification API je nezavisan od service workera i može se koristiti i bez njega
- e) Web-aplikacija ne može prikazati notifikaciju bez eksplicitne dozvole korisnika
- f) Push notifikacija neće biti primljena kao su ugašeni svi prozori preglednika

9. Neka su Milk, T i x definirani kao u nastavku

```
type Milk = {  
  brand : string;  
  fat: number;  
  volume: number;  
}  
  
type T = keyof Milk;
```

Označite ispravne naredbe ako je u postavkama TypeScriptovog prevodioca opcija strictNullChecks postavljena na false:

- a) let x : T = „brand“ | „fat“ | „volume“;
- b) let x : T = „brand“;
- c) let x : T;
- d) let x : T = „Milk“;
- e) let x : T = 1;

10. Ako su Milk i Beverage definirani kao u nastavku, što je Milk & Beverage?

```
type Milk = {  
  brand : string;  
  fat: number;  
  volume: number;  
}  
  
type Beverage = {  
  type : string;  
  brand : string;  
  volume : number;  
}
```

Odgovor: tip koji sadrži sljedeća svojstva

```
{  
  brand : string;  
  fat: number;  
  type : string;  
  volume: number;  
}
```

11. Prilikom **određivanja performansi web sjedišta** označite sve što je potrebno provesti:

- a) Procjenu troškova deploymenta
- b) Test opterećenja (load test)**
- c) Ispitivanje kapaciteta baze podataka (volume testing)**
- d) Ispitivanje responzivnosti (UX)
- e) Ispitivanje izdržljivosti (endurance testing)**
- f) Procjenu sigurnosnih ranjivosti

12. Ako je Milk definiran kao u nastavku, kako definirati Beverage koji za razliku od mlijeka nema fat, ali ima svojstvo type koje bi predstavljalo naziv vrste pića?

```
type Milk = {  
  brand : string;  
  fat: number;  
  volume: number;  
}
```

**a** `type Beverage = Milk - {"fat"} + {"type" : string};`

**b** `type Beverage = Partial<Milk, "fat"> & "type";`

**c** `type Beverage = (keyof Milk - {"fat"}) & {"type" : string};`

**d** `type Beverage = Omit<Milk, "fat"> & {"type" : string};`

**e** `type Beverage = Pick<Milk, "fat"> & {"type" : string};`

13. Označite sve ispravne programske odsječke

**a**

```
function pickRandomly(data : string) : string;  
function pickRandomly<T>(data : string | T[]) : string | T {  
  const pos = Math.floor(Math.random() * data.length);  
  return data[pos];  
}  
  
let s = "Web2";  
let c : string = pickRandomly(s);  
console.log(c);
```

**b**

```
function pickRandomly<T>(data : T[]) : T {  
  const pos = Math.floor(Math.random() * data.length);  
  return data[pos];  
}  
  
function pickRandomly(data : string) : string {  
  const pos = Math.floor(Math.random() * data.length);  
  return data[pos];  
}  
  
let s = "Web2";  
let c : string = pickRandomly(s);  
let a = [10, 20, 30, 40];  
let n : number = pickRandomly(a);
```

```
c
function pickRandomly<T>(data : T[]) : T;
function pickRandomly<T>(data : string | T[]) : string | T {
  const pos = Math.floor(Math.random() * data.length);
  return data[pos];
}

let a = [10, 20, 30, 40];
let n : number = pickRandomly(a);
console.log(n);

d
function pickRandomly(data : string) : string;
function pickRandomly<T>(data : T[]) : T;
function pickRandomly<T>(data : string | T[]) : string | T {
  const pos = Math.floor(Math.random() * data.length);
  return data[pos];
}

let s = "Web2";
let c : string = pickRandomly(s);
let a = [10, 20, 30, 40];
let n : number = pickRandomly(a);

e
function pickRandomly<T>(data : string | T[]) : string | T {
  const pos = Math.floor(Math.random() * data.length);
  return data[pos];
}

let s = "Web2";
let c : string = pickRandomly(s);
console.log(c);
```

14. Kod izrade **skalabilnih arhitektura aplikacija** označite sve tvrdnje u vezi horizontalnog skaliranja **koje su istinite**:
- a) Kod horizontalnog skaliranja osim scale out postoji i scale in jer kvalitetan skalabilan sustav trebao bi omogućiti i smanjenje kapaciteta
  - b) Horizontalnim skaliranjem povećavaju se performanse **postojećih** strojeva
  - c) Horizontalnim skaliranjem dodaje se **više** fizičkih strojeva ili resursa
  - d) Horizontalnim skaliranjem smanjuje se opterećenje na **svakom** stroju ili resursu
  - e) Horizontalnim skaliranjem povećava se opterećenje na **svakom** stroju
  - f) Kod horizontalnog skaliranja **ne postoji** scale in već samo scale out
15. Ako u argumentu funkcije izostavimo tip podatka, TypeScript će smatrati da je taj argument tipa any. Ako želimo da takav kod uzrokuje sintaksne pogreške prilikom prevođenja treba uključiti opciju:
- a) **noImplicitAny**
  - b) resolveJsonModule
  - c) noEmitOnError
  - d) declaration
  - e) strictNullChecks

16. Označite odgovor s ispravnim Typescript kodom koji bi omogućio da se sljedeći programski odsječak uspješno provede i ispiše 5.

```
let p : ImmutablePoint = new ImmutablePoint(2, 3);  
console.log(p.x + p.y);
```

**a**

```
class ImmutablePoint {  
  readonly x, y;  
  constructor(x: number, y:number) { }  
}
```

**b**

```
class ImmutablePoint {  
  public readonly x : number;  
  public readonly y : number;  
  constructor(x: number, y:number) { }  
}
```

**c**

```
class ImmutablePoint {  
  public readonly x : number;  
  public readonly y : number;  
  constructor(public readonly x: number, public readonly y:number) {  
    this.x = x;  
    this.y = y;  
  }  
}
```

**d**

```
class ImmutablePoint {  
  constructor(public readonly x: number, public readonly y:number) {}  
}
```

17. Što će se ispisati?

```
let obj = { broj: 8 };
let pObj = new Proxy(obj, {
  set: function (obj, prop, value) {
    if (prop === "broj") {
      if (value > 10) value = 10;
    }
    obj[prop] = value;
    return true; // Indicate success
  },
});
console.log(obj.broj);
console.log(pObj.broj);
obj.broj = 15;
console.log(obj.broj);
console.log(pObj.broj);
pObj.broj = 19;
console.log(obj.broj);
console.log(pObj.broj);
```

**Odgovor: 8 8 15 15 10 10**

18. Prednosti arhitekture u kojoj koristimo samo **jednostranične web-aplikacije** (u odnosu na druge dvije arhitekture) su:

- a) Lagano testiranje
- b) Jednostavnost
- c) Sadržaj se lako prilagodi klijentu
- d) Performanse (dio opterećenja se prebacuje na klijenta)
- e) Responzivnost (UX)
- f) Sigurnost



19. Neka je Milk definiran kao u nastavku.

```
type Milk = {  
  brand : string;  
  fat: number;  
  volume: number;  
}
```

Pretpostavimo da želimo napisati sljedeći kod

```
let milks : Milks = {};  
milks.DukatMali = {brand: "Dukat", volume: 0.5, fat : 3.2}  
milks.VindijskaVelika = {brand: "Vindijska", volume: 1.75, fat : 3.2}
```

Što od navedenog treba prethoditi takvom kodu?

**a** `type Milks = Milk[];`

**b** `interface Milks {  
 any:Milk  
};`

**c** `interface Milks {  
 [key:string] : Milk;  
};`

**d** `type Milks = {string : Milk}[];`

**e** `interface Milks {  
 [key:Milk]  
};`

20. Neka su u nekoj TypeScript datotecki zadani sučelje Point i funkcija distance kao u nastavku

```
interface Point {  
  x:number;  
  y:number;  
}  
function distance(p1:Point, p2: Point) : number {  
  return Math.sqrt((p1.x - p2.x) **2 + (p1.y - p2.y) ** 2);  
}
```

Označite sve programske odsječke koji su sintaksno ispravni.

**a** `let d = distance({x:1, y:2}, {x:4, y:6});`

**b** `const point1 = new Point(1, 2);  
const point2 = new Point(4, 6);  
let d = distance(point1, point2)`

**c** `const point1 : Point = {x: 1, y: 2};  
const point2 : Point = new Point(4, 6);  
let d = distance(point1, point2);`

**d** `const vector = {x : 1, y: 2, z : 3};  
const point = {x : 2, y: 1};  
let d = distance(point, vector);`

**e** `const vector = {x : 1, y: 2, z : 3};  
let d = distance(vector, vector);`

21. Označi sve točne tvrdnje:

- a) Kod u Typescriptu može generirati dodatni kod u Javascriptu, tj. 1 redak TS koda može uzrokovati nekoliko redaka JS koda
- b) Sintaksno neispravan kod u Typescriptu može se prevesti u JavaScript
- c) Svaki JavaScript kod je automatski sintaksno ispravni TypeScript kod
- d) Tipovi iz koda u TypeScripta se uklanjaju iz koda prevedenog u JavaScript
- e) TypeScript je nadogradnja JavaScripta u sintaksnom smislu (typed superset)

22. Ako su x, Milk i Beverage definirani kao u nastavku, označite sve naredbe koje su ispravne u metodi f:

```
type Milk = {  
  brand : string;  
  fat: number;  
  volume: number;  
}  
  
type Beverage = {  
  type : string;  
  brand : string;  
  volume : number;  
}  
  
function f(x : Milk & Beverage) : void {  
  ...  
}
```

& --> napravi uniju svojstava

- a) x.volume = 5.2;
- b) x.brand = „FER“;
- c) x.fat = 3;
- d) x.type = „Juice“;
- e) x.brand.push(„FER“);

23. Koje od sljedećih izjava su točne?

- a) Vue omogućuje da, ako želimo, komponente učitavamo i asinkrono
- b) Provide/inject mehanizam u Vueu nam omogućuje da proslijedimo svojstva niz stablo hijerarhije komponenti, čak i ako komponenta nije neposredni roditelj komponenti koja prima svojstva
- c) Kada napravimo produkcijski build aplikacije u Vue radnom okviru, sav javascript programski kod bude smješten u jednoj javascript datoteci
- d) Mutations u Vuexu su asinkrone
- e) U Vueu nema načina da komponenta dijete komunicira s roditeljem osim preko storea (npr. Vuex)
- f) Actions u Vuexu su asinkrone

24. Ako su x, Milk i Beverage definirani kao u nastavku, označite sve naredbe koje su ispravne u metodi f: (ko i 22 definicija osim f)

Function f(x : Milk | Beverage) : void {...} | --> radi presjek svojstava

- a) X.brand = „FER“;
- b) X.fat = 3;
- c) X.volume = 5.2;
- d) X.brand.push(„FER“);
- e) X.type = „Juice“;

25. Kod izrade **skalabilnih arhitektura aplikacija** označite sve tvrdnje u vezi **načela izrade dijagrama koje su istinite:**

- a) U načelu izrade dijagrama potiče se korištenje UML dijagrama, i to posebno obrazaca uporabe (use case), razreda (class) i komponenti (component)
- b) Dijagrami su samo zanimljive grafičke ilustracije i **ne** pomažu u shvaćanju dizajna sustava
- c) Ako se izrade na pravilan način dijagrami pomažu u shvaćanju dizajna sustava
- d) Namjena dijagrama je dokumentiranje sustava na jednoznačan način
- e) Namjena dijagrama je izrada grafički kvalitetnih ilustracija arhitekture sustava i zamjena opisa riječima sa živopisnim skicama
- f) Dijagrami se **ne** mogu koristiti za jednoznačnu komunikaciju 30 razvojnih inženjera ili između inženjera i korisnika sustava

26. Prednosti arhitekture u kojoj koristimo samo **statičke web-stranice** (u odnosu na druge dvije arhitekture) su:

- a) Performanse
- b) Interaktivnost
- c) Jednostavnost
- d) Sadržaj se lako prilagodi klijentu
- e) Sigurnost
- f) Responzivnost (UX)

27. Ako je u TypeScriptu napisan sljedeći kod:

```
function max(a, b) {  
    return a > b ? a : b;  
}  
...  
let m = max(3, 5);
```

Kojeg će tipa biti m? **Odgovor: ANY**

28. Kod izrade **skalabilnih arhitektura aplikacija** označite sve tvrdnje u vezi **načela visoke dostupnosti koje su istinite:**

- a) **Nije važno da li postoje neke greške unutar sustava koje klijentima nisu vidljive**
- b) Dovoljno je **jednom testirati** sustav na najvjerojatniji scenarij kvara i time će se ukloniti sve jedinstvene točke kvara sustava (single point of failure)
- c) **Sustav se smatra dostupnim sve dok obavlja svoje funkcije onako kako se očekuje iz perspektive klijenta**
- d) Važno je da li postoje neke greške unutar sustava koje **klijentima nisu vidljive**
- e) Sustav se smatra dostupnim sve dok obavlja svoje funkcije onako kako se očekuje iz perspektive **izvršitelja** i razvojnih inženjera, a pri tome perspektiva klijenta nije važna
- f) **Za poboljšanje otpornosti sustava potrebno je kontinuirano testirati različite scenarije kvarova i ukloniti jedinstvene točke kvara sustava (single point of failure)**

29. Kako treba deklarirati `z` da bi sljedeći programski odsječak bio ispravan u TypeScriptu?

```
z = 5;  
z = "Web2";
```

- a) **Let z : number | string;**
- b) `Const z: any;`
- c) `Let z : number & string;`
- d) `Let z = „“;`
- e) **Let z: any;**

30. Označite sve izjave u vezi particioniranja (sharding) koje su točne:

- a) **Postupak particioniranja podataka koristi se u slučaju da količina podataka za pohranu nadmašuje kapacitet sekundarne memorije („podaci ne stanu na disk“)**
- b) **Dva problema kod particioniranja podataka su kompleksni upiti u povezani upiti**
- c) U **svaki** master poslužitelj pohranjuju se **svi** podaci
- d) **Ako se koristi sharding podatkovni model potrebno je dizajnirati tako da ne zahtijeva kompleksne upite i joinove**
- e) Ako se koristi sharding onda se jedan master poslužitelj u principu **ne** zamjenjuje s više master poslužitelja
- f) Postupak particioniranja podataka **nikad** se ne kombinira s replikacijom

31. Što će se ispisati na izlazu nakon što sljedeći TypeScript prevedemo u JavaScript i pokrenemo?

```
let x : number | boolean = true;  
console.log(typeof(x));
```

Odgovor: **BOOLEAN**

## ZIR 2022

1. Ako je Milk definiran kao u nastavku, kako definirati Beverage koji za razliku od mlijeka nema fat i brand, ali ima svojstvo type koje bi predstavljalo naziv vrste pića?

```
type Milk = {  
  brand : string;  
  fat: number;  
  volume: number;  
}
```

a

```
type Beverage = (keyof Milk - {"fat" | "brand"}) & {"type" : string};
```

b

```
type Beverage = Pick<Milk, Omit<"fat" | "brand">> & "type";
```

c

```
type Beverage = Pick<Milk, "volume"> & {"type" : string};
```

d

```
type Beverage = Omit<Milk, "fat"> & {"type" : string};
```

e

```
type Beverage = Milk - {"fat" | "brand"} + {"type" : string};
```

2. Ako je u Typescriptu napisan sljedeći kod:

```
function demo(a:number, ...b:number[]) {  
  return [a, b];  
}
```

Kojeg tipa je povratna vrijednost iz funkcije demo?

a) any

b) (number | number[])[]

c) number[]

d) number | number[]

e) any[]

return [a, b]; vraća array unutar arraya , [1,[2,3]]

return [a,...b] vraća array [1,2,3]

3. Zamislimo hipotetski primjer u kojem Edgar i ISVU razmjenjuju podatke o studentima, npr. Tako da Edgar preuzima popis upisanih studenata na predmetu koristeći za to predviđeni servis. Takav servis nije javan, već se pravo pristupa određuje koristeći OAuth2/OIDC protokole. Koji OAuth2/OIDC tok bi se koristio u ovom primjeru?
- a) Trusted Application Flow
  - b) Authorization code Flow + PKCE
  - c) Resource Owner Password Flow
  - d) Client Credentials Flow**
  - e) Implicit Flow
4. Koji isječak koda je potrebno upisati u HTML5 dokument umjesto xxxx da bi se u pregledniku ispravno koristio Web Storage API?

```
<!DOCTYPE html>
<html>
<head>
<script>
function clickCounter() {
    if (typeof(Storage) !== "undefined") {
        if (XXXXXXXXXXXX.clickcount) {
            XXXXXXXXXX.clickcount = Number(XXXXXXXXXXXX.clickcount)+1;
        } else {
            XXXXXXXXXX.clickcount = 1;
        }
        document.getElementById("result").innerHTML = "Kliknuli ste na tipku " +
XXXXXXXXXXXX.clickcount + " puta u ovoj sesiji.";
    } else {
        document.getElementById("result").innerHTML = "Nažalost vaš preglednik ne podržava HTML Web
Storage API.";
    }
}
</script>
</head>
<body>

<p><button onclick="clickCounter()" type="button">Kliknite ovu tipku</button></p>
<div id="result"></div>
<p>Kliknite tipku kako biste vidjeli povećanje brojača.</p>
<p>Zatvorite karticu preglednika (ili prozor) i pokušajte ponovno, a brojač će se resetirati.</p>

</body>
</html>
```

- a) sessionStorage**
- b) navigator.storage
- c) window.storage
- d) document.storage
- e) session

5. Koje od sljedeći tvrdnji vezane uz PWA tehnologije su istinite?
- a) Service worker nema pristup DOM-u
  - b) Service worker radi samo ako je stranica poslužena putem HTTPS-a ili preko localhosta
  - c) Service worker se izvodi mimo glavne preglednikove UI dretve
  - d) Service worker ne može koristiti sinkroni localStorage API, pa zato tipično koristi asinkroni IndexedDB
  - e) Notification API je nezavisan od service workera i može se koristiti i bez njega
  - f) VAPID ključ nam treba kako bi od korisnika zatražili dozvolu za korištenje push notifikacija
6. JSON kojeg OAuth2/OIDC server vrati na pristupnoj točki /.well-known/openid-configuration između ostalog sadrži i vrijednost zavedenu pod jwks\_uri. Ta vrijednost služi
- a) Za informaciju o adresi pristupne točke koja će vratiti detaljne informacije o korisniku, tj. vlasniku nekog tokena
  - b) Za informaciju o adresi na kojoj se mogu preuzeti javni ključevi kojima se provjerava token
  - c) Za registraciju aplikacije koja će se spojiti na navedeni OIDC server
  - d) Za informaciju o popisu podržanih opsega
  - e) Za dohvat popisa svih registriranih korisnika

7. Koji isječak koda u HTML5 dokument umjesto xxxx da bi web worker api radio ispravno?

```
<!DOCTYPE html>
<html>
<body>

<p>Brojač: <output id="result"></output></p>
<button onclick="startWorker()">Pokreni Worker</button>
<button onclick="stopWorker()">Zaustavi Worker</button>

<script>
var w;

function startWorker() {
  if(typeof(Worker) !== "undefined") {
    if(typeof(w) == "undefined") {
      w = new Worker("demo_workers.js");
    }
    XXXXXXXXXXXXXXXX {
      document.getElementById("result").innerHTML = event.data;
    };
  } else {
    document.getElementById("result").innerHTML = "Nažalost vaš preglednik ne podržava HTML Web Workers API.";
  }
}

function stopWorker() {
  w.terminate();
  w = undefined;
}
</script>

</body>
</html>
```

Kliknite ovu tipku

Kliknuli ste na tipku 3 puta u ovoj sesiji.

Kliknite tipku kako biste vidjeli povećanje brojača.

Zatvorite karticu preglednika (ili prozor) i pokušajte ponovno, a brojač će se resetirati.

**Odgovor: w.onMessage = function(event)**



8. Unesi CSS koji će nastati prevođenjem SAAS koda:

```
%default {  
  padding: 1rem;  
}  
$primary-color: blue;  
$secondary-color: red;  
  
header {  
  @extend %default;  
  p {  
    font-size: 2rem;  
    color: $primary-color;  
  }  
}
```

Odgovor: header {  
padding: 1rem;  
}  
header p {  
font-size: 2rem;  
color: blue;  
}

9. Ako su MemoryCard i Phone definirani kao u nastavku, odaberite naredbu kojom ćemo u metodi f ispisati operacijski sustav mobitela (svojstvo OS u Phone)

```
type MemoryCard = {  
  name: string;  
  storage : {  
    capacity : number;  
    writeSpeed : number;  
    readSpeed : number;  
  }  
}  
  
type Phone = {  
  name: string;  
  storage : {  
    capacity: number;  
    type: string;  
  }  
  os: string;  
}  
  
function f(x : MemoryCard | Phone) : void {  
  ...  
}
```

- a) Console.log(x.os ?? "");
- b) If(typeof x === typeof Phone) console.log(x.os);
- c) If(x instanceof Phone) console.log(x.os);
- d) If(x.os !== undefined) console.log(x.os);
- e) If("os" in x) console.log(x.os);

10. Označite sve postupke kojima se otklanja sigurnosni nedostatak kada prilikom učitavanja XML datoteka sličnim niže navedenoj dolazi do **preljeva međuspremnika (buffer overflow) na poslužitelju?**

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE foo [
  <!ELEMENT foo ANY>
  <!ENTITY bar "World ">
  <!ENTITY t1 "&bar;&bar;">
  <!ENTITY t2 "&t1;&t1;&t1;&t1;">
  <!ENTITY t3 "&t2;&t2;&t2;&t2;&t2;">
]>
<foo>
  Hello &t3;
</foo>
```

- a) Izbjegavati korištenje složenijih XML struktura ako nije baš potrebno  
b) Dodati neku tajnu (token ) uz Session ID, a ne prihvaćati sve XML datoteke automatski  
c) Napraviti validaciju / sanitizaciju XML dokumenata prije parsiranja  
d) Uvijek i potpuno onemogućiti unos svih XML datoteka  
e) Filtriranje dozvoljenih IP adresa i raspona IP adresa
11. Ako zlonamjerni napadač vidi da ima pristup putanji <https://www.insecure-web-server-example.com/user?id=10023> te zbog nesigurnih referenci na objekte na web poslužitelju može pristupiti i putanji <https://www.insecure-web-server-example.com/user?id=10024>, koji je to nedostatak? **Odgovor: LOŠA KONTROLA PRISTUPA**
12. Prilikom prijave na sustav e-Građani možete se prijaviti tokenom koje izdaje banke u Hrvatskoj. U kontekstu vanjske usluge za prijavu, što ste vi kao osoba koja se prijavljuje u sustav e-Građani? **Odgovor: RESOURCE OWNER**
13. Kojoj metodologiji odnosno radnom okviru odgovara sljedeći kod:

```
<figure class="md:flex bg-slate-100 rounded-xl p-8 md:p-0 dark:bg-slate-800">
  
  <div class="pt-6 md:p-8 text-center md:text-left space-y-4">
    <blockquote>
      <p class="text-lg font-medium">
        Lorem ipsum...
      </p>
    </blockquote>
  </div>
</figure>
```

**Odgovor: Utility classes CSS radni okviri poput Tailwinda ili Tachyonsa**

14. Okvir (frame) je najmanji element komunikacije HTTP/2 protokolom. Koje informacije okvir sadržava? **Odgovor: ZAGLAVLJE OKVIRA (FRAME HEADER) I IDENTIFIKATOR TOKA (ID) KOJEMU OKVIR PRIPADA**

15. Pretpostavimo da na Edgaru postoji web-servis koji temeljem tokena kojeg izdaje AAI@EduHr prepoznaje korisnika te mu vraća popis svih njegovih testova i bodova. Želite napisati aplikaciju za pametni TV koja će se svakih sat vremena spojiti na taj servis i provjeriti ima li novih podataka, ali ne želite se interaktivno prijavljivati na TV-u, već svoje korisničko ime i lozinku za [AAI@Edu.HR](mailto:AAI@Edu.HR) pohranite u konfiguracijskoj datoteci vaše aplikacije. Koji OAuth2/OIDC tok se koristi između vaše aplikacije i servisa [AAI@Edu.HR](mailto:AAI@Edu.HR) u ovom primjeru? **Odgovor: RESOURCE OWNER PASSWORD FLOW**

16. Kod određivanja prioriteta HTTP/2 tokova (stream prioritization) kako se izgrađuje stablo prioriteta (priorization tree)? **Odgovor: PROPORCIONALNO UDALJENOSTI TOKA OD ČVORA STABLA PRIORITETA I DODIJELJENOJ VRIJEDNOSTI PRIORITETA**

17. Što će se ispisati na izlazu nakon što sljedeći TypeScript prevedemo u JavaScript i pokrenemo?

```
let x : number | {b : boolean} = 5;  
console.log(typeof(x));
```

**Odgovor: NUMBER**

18. Ako je u TypeScriptu napisan kod:

```
function demo(a:string, ...b:number[]) {  
    return [a, ...b];  
}
```

Kojeg tipa je povratna vrijednost iz funkcije demo?

**Odgovor: (string | number)[ ]**

19. Prilikom prijave na sustav e-Građani možete se prijaviti tokenom koje izdaje banke u Hrvatskoj. U kontekstu vanjske usluge za prijavu, banka koja je izdala token je? **Odgovor: AUTHORIZATION SERVER**

20.

Neka je `Milk` definiran kao u nastavku.

```
type Milk = {  
  brand : string;  
  fat: number;  
  volume: number;  
}
```

Pretpostavimo da želimo napisati sljedeći kod

```
let milks : Milks = {};  
let DukatMali = {brand: "Dukat", volume: 0.5, fat : 3.2};  
let VindijaVelika = {brand: "Vindija", volume: 1.75, fat : 3.2};  
milks.DukatMali = 7;  
milks.VindijaVelika = 13;
```

Što od navedenog treba prethoditi takvom kodu?

**a**

```
type Milks = number[];
```

**b**

```
interface Milks {  
  [key:string] : Milk;  
};
```

**c**

```
interface Milks {  
  [key:string] : number  
};
```

21. Kojoj metodologiji tj radnom okviru odgovara sljedeći kod:

```
<style>  
  .photo { }  
  .photo__ing { }  
  .photo__caption { }  
</style>  
  
<figure class="photo">  
    
  <figcaption class="photo__caption">Look at me!</figcaption>  
</figure>
```

Odgovor: BEM

22. Što će se ispisati?

```
let obj = {
  x: 1,
  y: 2
};
let pObj = new Proxy(obj, {
  get: function (target, prop, receiver) {
    if (prop === "x") {
      return "x is private";
    }
    return Reflect.get(...arguments);
  },
  set: function (obj, prop, value) {
    if (prop !== "y") {
      obj[prop] = value;
    }
    return true;
  },
});
console.log(pObj.x);
console.log(pObj.y);
obj.x = 17;
obj.y = 17;
console.log(pObj.x);
console.log(pObj.y);
pObj.x = 28;
pObj.y = 28;
console.log(pObj.x);
console.log(pObj.y);
// -----
console.log(obj.x);
console.log(obj.y);
```

**Odgovor: x is private 2 x is private 17 x is private 17 28 17**